# DIGITAL PHASE DISCRIMINATION BASED ON FREQUENCY SAMPLING

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

The present invention relates to digital phase discrimination.

### 2. State of the Art

Phase discrimination is important in digital radio communications, in particular in any angle modulation digital radio receiver. Phase discrimination and frequency discrimination are closely related. Frequency discrimination is typically performed using analog circuitry, e.g. an IQ frequency discriminator. Analog frequency discriminators have substantial drawbacks. In the case of an IQ frequency discriminator, the discriminator requires a number of analog components, two A/D conversions and a numerical arctangent operation, rendering the circuit quite complex.

Known methods exist for producing a value representing the instantaneous phase of a signal using only digital logic elements. Various such methods are described in U.S. Patent 5,084,669, incorporated herein by reference. In particular, the foregoing patent describes a digital circuit for determining the instantaneous phase of a signal, from which the instantaneous frequency may be obtained if desired. Although the implementation of the circuit is all digital, it is quite involved. An improved method and apparatus for determining in a simple, all-digital manner the instantaneous phase of a signal would therefore likely be well-received by those skilled in the art.

## SUMMARY OF THE INVENTION

The present invention, generally speaking, provides a simple, all-digital method and apparatus for determining the phase of a first clock signal relative to a second clock signal. The first clock signal may be a digital approximation of a periodic analog signal such as an RF signal. A sampling technique is

employed that produces a stream of digital bits containing relative phase information. From the stream of digital bits is formed a digital word indicative of the relative phase. The digital word may be formed using a digital filter. Advantageously, an extensive body of digital filtering techniques applicable to Sigma-Delta (sometimes referred to as Delta-Sigma) A/D converters may be applied directly to the digital stream. By using an appropriately-chosen weighting function, high accuracy may be obtained.

## BRIEF DESCRIPTION OF THE DRAWING

The present invention may be further understood from the following description in conjunction with the appended drawing. In the drawing:

Figure 1 is a block diagram illustrating a sampled-data model of a Sigma-Delta modulator and of a sampling circuit applied to frequency sampling in accordance with one embodiment of the present invention;

Figure 2 is a table helpful in explaining operation of the circuit model of Figure 1 in the instance of an input frequency that is 0.6875 times a reference frequency;

Figure 3 is a timing diagram illustrating the principle of operation of the circuit model of Figure 1 as applied to frequency sampling;

Figure 4 is a schematic diagram of one example of a frequency sampling circuit described by the circuit model of Figure 1;

Figure 5 is a first timing diagram illustrating operation of the frequency sampling circuit of Figure 4;

Figure 6 is a second timing diagram illustrating operation of the frequency sampling circuit of Figure 4;

Figure 7 is a graph of two alternative weighting functions that may be used to perform digital filtering of a digital bit stream produced by a circuit such as that of Figure 4;

Figure 8 is a graph illustrating the accuracy obtained from a digital frequency discriminator using a constant weighting function;

Figure 9 is a graph illustrating the accuracy obtained from a digital frequency discriminator using a triangular weighting function; and

Figure 10 is a block diagram of one example of a digital filter that may be used in conjunction with a frequency sampling circuit such as that of Figure 4.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The approach followed by the digital frequency discriminator of the present invention may be appreciated by analogy to Sigma-Delta A/D conversion, well-documented in the prior art by such references as "Oversampling Delta-Sigma Data Converters", Candy, et al., *IEEE Press*, pages 1-6, Piscataway, NJ (1992). A Sigma-Delta converter modulates a varying-amplitude analog input signal into a simple digital code at a frequency much higher than the Nyquist rate. The design of the modulator allows resolution in time to be traded for resolution in amplitude. A sampled-data circuit model of a Sigma-Delta modulator, shown in Figure 1, may be directly applied to frequency sampling as described herein.

Referring to Figure 1, an input signal $x_i$ occurring at sample time i has subtracted from it the output signal $y_i$ at sample time i. The result is applied to an accumulator having an output signal $w_i$. A "new" input signal of the accumulator at sample time i is combined with the "old" output signal of the accumulator to form a new output signal of the accumulator. The output signal of the accumulator is quantized, the quantization being represented as the addition of an error $e_i$. The output signal of the quantizer is the final output signal $y_i$.

Assume now that $x_i$ is the ratio of two frequencies and that the quantizer is a two-level quantizer. Further assume that the ratio of the two frequencies for the time period in question is, say, 0.6875. As shown in Figure 2, the latter value is accumulated a first time, giving an accumulated value of 0.6875. This valuing being less than 1, the value 0.6875 is again added to the accumulated value, giving a new accumulated value of 1.375. Since this value

is now greater than 1, 1 is subtracted from 0.6875 and the result (0.6875 - 1 = -0.3125) added to the accumulator to give a value of 1.0625. Operation proceeds in this fashion. During the foregoing sequence of operations, a data stream is produced by taking the integer portion, 1 or 0, of each accumulated value.

Referring to Figure 3, the interpretation of the sequence of numbers shown in Figure 2 may be appreciated. Two clock signals are shown. Again, it is assumed that the ratio of the frequency of the upper clock signal to that of the lower clock signal during the period of interest is 0.6875. At time t = 0, rising edges of both clock signals coincide. At the first subsequent rising edge of the lower clock signal, 0.6875 periods of the upper clock signal have elapsed. At the next rising edge of the lower clock signal, 1.375 periods of the upper clock signal have elapsed. At the next rising edge of the lower clock signal, 1.0625 periods of the upper clock signal have elapsed since the elapse of the first period of the upper clock signal, and so on.

A schematic diagram of a capture circuit, or frequency sampling circuit, that may be used to data samples corresponding to the data stream described in the foregoing example is shown in Figure 4. In the illustrated embodiment, it is assumed that the ratio of the clock signals is such that no more than one rising edge of the faster clock will occur during a single period of the slower clock. In other embodiments, this assumption need not apply.

The capture circuit includes a input portion 401 and an output portion 403. The input portion includes two sections Ch1 and Ch2 that must be carefully matched to minimize errors. Each section comprises a chain of two or more D flip-flops coupled in series. In the following description, the same reference numerals will be used to reference the respective flip-flops themselves and their respective output signals.

Within each section, the first flip-flop in the chain is clocked by a sampled clock signal Fx. The succeeding flip-flops in the chain are clocked by a sampling clock signal Fs. The D input of the first flip-flop Q1 in the upper

section is coupled to the $\overline{Q}$ output of the same. The D input of the first flip-flop in the lower section is coupled to the Q output of the first flip-flop in the upper section. The remaining flip-flops in both sections are coupled in series--i.e., Q to D, Q to D.

The function of the input portion is to 1) produce two signals, logical inverses of one another, that transition on rising edges of the clock signal Fx; 2) to latch the values of the two signals on the rising edge of the clock signal Fs; and 3) to detect transitions from one clock to the next. Additional intermediate stages in series with Q3 and Q4 may be required to minimize metastability resulting from the asynchrony of the two clock signals, and in fact multiple such stages may be desirable in a particular design.

The output portions include, in an exemplary embodiment, three two-input NAND gates. Respective NAND gates N1 and N2 are coupled to the D and $\overline{Q}$ signal of the final flip-flop stages of the input sections. Output signals of the NAND gates N1 and N2 are combined in the further NAND gate N3 to form the final output of the capture circuit.

The function of the output portion is to detect a change in the input clock signal level from one sample clock to the next in either of two channels formed by the two input sections. The two input sections function in a ping-pong fashion, alternately detecting changes in the input clock signal level.

Operation of the capture circuit of Figure 4 may be more fully appreciated with reference to the timing diagram of Figure 5. The first stages of the two channels form inverse signals Q1 and Q2 approximately coincident with (but slightly delayed from) rising edges of the input clock signal. The signals Q3 and Q4 are formed by sampling the signals Q1 and Q2, respectively, in accordance with the sample clock. The signals Q5 and Q6, respectively, are delayed replicas of the signals Q3 and Q4. The NAND gates together realize the logic function $X = Q3 \cdot \overline{Q5} \vee Q4 \cdot \overline{Q6}$.

In the example of Figure 5, the illustrated signals are all idealized square-wave signals. In actuality, the signals will have finite rise and fall times. The possible effect of the finite rise and fall times of the signals Q1 and Q2 and the asynchrony of the circuit is metastability, as illustrated in Figure 6. Here, the signals Q3 and Q5 and the signals Q4 and Q6 are each in an indeterminate state for one cycle. The resulting output of the circuit may or may not be correct. However, because the decision was a "close call" to begin with, the effect of an occasional erroneous decision on the overall operation of the circuit is negligible. The time window of instability is reduced by increasing the overall gain in the path. If the gain in Q3 and Q9 is sufficient to reduce the probability of an error to an acceptable level, then no additional circuitry is required. If not, then additional circuitry will be required to increase the gain.

In order to recover the ratio of the frequencies of the two clock signals from the data stream produced by a capture circuit such as the one of Figure 4, digital filtering is applied. Advantageously, an extensive body of digital filtering techniques applicable to Sigma-Delta (or Delta-Sigma) A/D converters may be applied directly to the digital stream. Furthermore, by using an appropriately-chosen weighting function, high accuracy may be obtained.

The weighted sum of products is an example of an FIR filter. The weighting function described heretofore is therefore that of an FIR filter in digital filtering theory. It should be recognized, however, that IIR filters can also be used. In the process of FIR digital filtering, the weighting function is applied to a "window" of data samples to obtain an estimate of the ratio of frequencies in the center of the window. The window is then "picked up and moved" to the next sequence of samples. Windowing will typically overlap. A window may include 256 samples, for example.

Referring to Figure 7, two alternative weighting functions are shown for a window of 256 samples. The weighting functions are normalized, meaning that the area under the weighting function is unity. One weighting function,

indicated in dashed lines, is a straight-line, constant weighting function. Another weighting function, indicated in solid line, is a triangular weighting function. The weighting function is the impulse response function in digital filters.

Results of digital filtering using the straight-line weighting function and the triangular weighting function respectively, are shown in Figure 8 and Figure 9. In the case of both Figure 8 and Figure 9, the frequency ratio was increased from just under 0.687 to just over 0.693. As seen in Figure 8, using a straight-line weighting function, the quantized signal oscillates between two levels that are adjacent to the input in such a manner that its local average equals the average input. The average error was calculated to be 1772ppm. As seen in Figure 9, using a triangular weighting function, the quantized signal tracks the input with an average error of 83ppm.

A schematic diagram of an exemplary frequency accumulator that applies a triangular weighting function and that may be used to accomplish the desired digital filtering is shown in Figure 10. In the example shown, the frequency accumulator uses a 7-bit counter 101, a 14-bit adder 103 and a 14-bit register 105. The 7-bit counter is clocked by the sample frequency Fs. The output of the 7-bit counter is provided to one input of the adder. The function of the 7-bit counter is to count up from 0 to 127 and then down from 127 to 0. The count of 127 occurs twice in succession. This behavior is achieved using a flip-flop 107. The flip-flop is clocked by the sample frequency Fs. A Terminal Count signal of the 7-bit adder is input to the flip-flop. The output of the flip-flop is coupled to a Count Down input of the 7-bit counter.

The "oversampled" data stream is coupled to a control input of the adder. When the current bit of the data stream is a 1, an addition is performed. When the current bit is a 0, no addition is performed. A Carry In input of the adder is tied high, effectively causing the range of weights to be 1 to 128.

The 14-bit register is clocked by the sample frequency Fs. Its output is applied to the other input of the adder. Its input receives the output word

produced by the adder. The function of the 14-bit adder is to perform an accumulation operation for 256 clocks. At the conclusion of the 256 clocks, the output of the 14-bit adder is used as an estimator for the frequency ratio. More particularly, in the example shown, the output of the accumulator is equal to R x 128 x 129, where R is the frequency ratio estimator.

The foregoing technique may be readily extended to phase discrimination. Various different methods and apparatus for digital phase discrimination will be described entailing different design tradeoffs.

The first method is conceptually straightforward but computationally expensive. Referring to Figure 11A, the same observed frequency data stream and the same set of weights corresponding to a triangular weighting function (Fig. 11B) are used. The ratio of the reference frequency to the sampled frequency over a relatively long period of time is first determined using the technique described previously. Having obtained this frequency ratio estimator, short-term frequency deviations are estimated by calculating the same frequency estimate as before but at a relatively high rate, as often as once per sample period. That is, successive samples are all taken using the circuit of Figure 10, as often as each sample period. The difference ($\Delta F$) of each frequency estimate (F) from the previously-determined frequency ratio (Fr) is calculated, multiplied by an appropriate scale factor k and accumulated to obtain a corresponding phase estimate Pf. (The first value of Pf is an arbitrarily chosen initial condition, chosen for comparison to an ideal estimate. In practice, the phase may be initialized to a value based on a priori knowledge of signal characteristics, or, absent such a priori knowledge, may be set to zero upon detection of a phase inflection point.)

A phase-plot simulation comparing actual phase of a specified waveform (solid-line) with estimated phase using the foregoing phase estimation method (dashed-line) is shown in Fig. 11C.

The foregoing "frequency difference" phase estimation method is computationally expensive because of the need to calculate frequency estimates

at a relatively high rate. A "pre-summation difference" phase estimation method obviates this requirement. Referring to Figure 12A, instead of subtracting the frequency ratio from a frequency estimate, the frequency ratio Fr is subtracted from the sampled data stream itself. Assuming that the data stream is a bit stream of ones and zeros only, and assuming a frequency ratio Fr = 0.6875, then the pre-summation difference Y will have one of only two values, Y = 1 - 0.6875 = .3125 or Y = 0 - 0.6875 = - 0.6875. The Y values are accumulated to obtain corresponding values PX. Phase estimates $PP_n$ are obtained by filtering the PX values in substantially the same manner as described previously in relation to forming frequency estimates (using the identical weighting function, Fig. 12B, for example) with the exception that the filtered values are scaled by the scale-factor k.

The pre-summation difference phase-calculation may be shown to be mathematically equivalent to the frequency difference phase calculation. Simulation results, shown in Fig. 12C, are therefore the same as in Fig. 11C. The hardware realization, however, may be considerably simpler using the pre-summation difference phase calculation, since only one computation is required per phase point. Such a hardware realization is shown in Figure 13.

The pre-summation difference phase estimator of Figure 13 includes generally a first accumulator ACC1, a weight generator WG similar or identical to the weight generator previously described in relation to Figure 10, and a second accumulator ACC2.

The accumulator ACC1 functions to produce phase numbers $PX_i$ in correspondence to bits (or in other embodiments, symbols) of the observed frequency data stream and includes a multiplexer 1301, an adder 1303 and a register (e.g., a 16-bit register) 1305. The multiplexer 1301 selects one of the two possible values of $Y_i$ in accordance with the value of X and applies $Y_i$ to the adder 1303. The register value is added to $Y_i$ to form $PX_i$, which is then strobed into the register. The adder 1303 and register 1305 therefore accumulate the $PX_i$ values.

The $PX_i$ values are then filtered in the accumulator ACC2, which includes a multiplier 1307, an adder 1309 and a register 1311. The multiplier receives weights from the weight generator WG and $PX_i$ values from the accumulator ACC1. Respective weights and $PX_i$ values are multiplied and the products accumulated, e.g., for 128 clock cycles, to produce a phase estimator PP. The multiplier may be constructed so as to apply the scale factor k to each product during the accumulation process.

An even simpler realization may be achieved using an integer difference phase calculation. The integer difference phase calculation is not mathematically equivalent to the foregoing methods, but is very close. Referring to Fig. 14A, this method uses, in addition to the observed frequency data stream, a reference frequency data stream that would result if the reference frequency were applied to the capture circuit of Figure 4 (with the same clock). A running sum $D_i$ is then formed of the integer difference $X_i$-$R_i$. In many practical applications, such as the one illustrated in Figures 14 and 15, $D_i$ will have the values 1, 0 and -1 exclusively. The general case in which $D_i$ takes on other values may be appreciated and understood, however, from the present example, and is embraced by the present description.

Phase estimates are formed by filtering the $D_i$ values in the same or similar manner as previously described. The same triangular weighting function may be used Fig. 14B. The integer difference phase calculation method produces identical simulation results, Fig. 14C, as the preceding methods.

Referring to Figure 15, in the instance where D takes on the values 1, 0 and -1 exclusively, the corresponding hardware realization may be substantially simplified (as compared to that of Figure 13, for example).

The integer difference phase estimator of Fig. 15, like that of Fig. 13, includes generally a first accumulator ACC1, a weight generator WG, and a second accumulator ACC2. The accumulator ACC1 is of considerably different construction than the corresponding structure of Figure 13. The accumulator

ACC1 of Figure 15 includes a reference pattern generator 1501, a 1-bit subtractor 1503, a 2-bit adder 1505 and a 2-bit register 1507. The 1-bit subtractor subtracts respective R values from respective X values. The 2-bit adder and the register accumulate the resulting $D_i$ values which, as explained previously, may be constrained to 1, 0, -1 only.

The weight generator WG and the accumulator ACC2 are substantially the same as in Figure 13, described previously. However, because $D_i$ takes on the values 1, 0 and -1 exclusively, no multiplier is required. Instead, if $D_i$ = 1, the weight value is added to the accumulated value, and if $D_i$ = -1, the weight value is subtracted. (If $D_i$ = 0, the accumulated value remains unchanged.) The savings of a hardware multiplier is a particular advantage of the implementation of Fig. 15.

A further method of phase estimation is referred to as the clock measure phase calculation method. Referring to Figure 16A, this method is similar to the previous integer difference phase calculation method insofar as R, X and D are concerned. This method, however, uses in addition to the reference frequency data stream R, "clock measure" numbers RG, which are the same as the numbers appearing in Fig. 2. Moreover, the weight function used is distinctly different, as shown in Figure 16B. Clock measure phase estimate values PC are obtained using the following formula:

$$PC_n = k \cdot (D_n - frac(RG_n) + 0.5 + \sum_i (W_i \cdot X_{i+n-64}))$$

Simulation results using the clock measure phase calculation method are shown in Figure 16C.

Referring to Figure 17, the clock measure phase estimator includes generally a first accumulator ACC1, a weight generator WG, and a second accumulator ACC2. The estimator additionally includes a summation block 1701.

The accumulator block ACC1 is substantially the same as the accumulator block ACC1 of Figure 15. Note, however, that the reference pattern generator generates both the reference frequency data stream R, used within the accumulator ACC1, and the clock measure data stream RG which is input to the summation block 1701.

The weight generator includes a counter 1703 and weight generator logic 1705.

The accumulator ACC2 includes an adder 1707 and a register 1709. When X = 1, the weight value from the weight generator is added to the contents of the register 1709. The output of the adder becomes the new input of the register, which performs an accumulator operation for, e.g., 128 clock cycles.

At the conclusion of the accumulation operation of ACC2, the outputs of ACC1 and ACC2, together with the corresponding RG value, are summed in the summation block 1701.

It will be appreciated by those of ordinary skill in the art that the invention can be embodied in other specific forms without departing from the spirit or essential character thereof. The presently disclosed embodiments are therefore considered in all respects to be illustrative and not restrictive. The scope of the invention is indicated by the appended claims rather than the foregoing description, and all changes which come within the meaning and range of equivalents thereof are intended to be embraced therein.